Multi-robot Dynamic Coverage of a Planar Bounded Environment

Maxim A. Batalin Gaurav S. Sukhatme

Robotic Embedded Systems Laboratory, Robotics Research Laboratory, Computer Science Department
University of Southern California
Los Angeles, CA, USA
maxim@robotics.usc.edu, gaurav@usc.edu

Abstract

The traditional approach to measure the efficiency of a (static) coverage task is the ratio of the intersection of the areas covered by sensors, to the total free space in the environment. Here we address the dynamic coverage problem, which requires all areas of free space in the environment to be covered by sensors in as short a time as possible. We introduce a frequency coverage metric *that measures the frequency of every-point coverage, and propose a decentralized algorithm that utilizes locally available information about the environment to address this problem. Our algorithm produces exploratory,* patrol-like *behavior. Robots deploy communication beacons into the environment to mark previously visited areas. These nodes act as local signposts for robots which subsequently return to their vicinity. By deploying such (stationary) nodes into the environment robots can make local decisions about their motion strategy. We analyze the proposed algorithm and compare it with a baseline approach - a modified version of a static coverage algorithm described in [1].*

1 Introduction

Coverage [2] is the problem of arranging sensors in the environment, usually with the aim of detecting targets. The target-detection sensors are often mounted on mobile robots, thereby reducing the problem to one of robot positioning. Such a capability is of obvious use in the detection of unfriendly targets (e.g military operations), monitoring (e.g. security), or urban search and rescue (USAR) in the aftermath of a natural or man-made disaster (e.g. building rubble due to an earthquake or other causes).

We are particularly interested in the regime where the number of robots is such that no static assignment of robots to vantage points, guarantees full coverage of the environment. This will often be the case in large environments. Further, in some applications coverage of every point of the environment at every instant may not be required. Thus we constrain the problem to the case where every point of the environment should be covered with at least a certain frequency. While we assume that coverage is necessary to enable capabilities such as target tracking, however we do not address those problems here. We focus on the problem of positioning multiple robots in a planar bounded environment to maximize their sensor coverage over time.

Our solution to the problem relies on the deployment of beacons into the environment as support infrastructure which the robots use to solve the coverage problem. Robots explore the environment, and based on certain local criteria, drop a beacon into the environment, from time to time. Each beacon is equipped with a small processor, a radio of limited range, and a compass. We describe a decentralized algorithm that performs the coverage task successfully using only local sensing and local interactions between the robots and beacons. The fundamental constraint that we impose on the solution is the lack of global information about the environment (neither a map nor an access to global positioning information). We introduce a *frequency coverage metric* that measures frequency of every-point coverage.

We compared our algorithm to a (modified) version of a prior approach in which robots disperse themselves in the environment by locally repelling interaction in order to 'fan out' to expand coverage (we call this the *Molecular* approach [1]). Simulations show that our algorithm outperforms the molecular algorithms. In addition, the present algorithm deploys a static network of nodes into the environment which has applications other than coverage.

2 Related Work

The coverage paradigm was formulated by Gage [2] and divided into three groups of useful behaviors. Blanket (or Field) coverage, that aims to achieve a static arrangement of agents to maximize the detec-

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **2004**	2. REPORT TYPE	3. DATES COVERED **00-00-2004 to 00-00-2004**

4. TITLE AND SUBTITLE **Multi-robot Dynamic Coverage of a Planar Bounded Environment**	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of Southern California,Computer Science Department,Los Angeles,CA,90089-0781**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S) 11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**6**	

tion rate of the targets in the sensor shadows. Barrier coverage, whose objective is to achieve a static arrangement of agents with the task of minimizing the probability of undetected target penetration through the barrier. Sweep coverage, that essentially represents a moving barrier coverage or can be achieved using random uncoordinated motion of agents (as shown in [2]). The problem discussed here is closely related to the sweep coverage problem.

The problem of dynamic coverage is also related to the exploration problem in an unknown environment which has been studied by several authors [3, 4, 5]. The frontier-based approach, described in detail in [3, 4], concerns itself with incrementally constructing a global occupancy map of the environment. The map is analyzed to locate the 'frontiers' between the free and unknown space. Exploration proceeds in the direction of the closest frontier. The multi-robot version of the same problem was addressed in [6]. [7] discusses the problem of deployment of distributed sensors (robots) in the wireless adhoc network domain. In their setup, the communication ranges between the robots are assumed to be limited and the environment is assumed to be big enough so that the network connectivity cannot be maintained. A random-walk algorithm is used to disperse the robot network into the environment to support communication.

The approach proposed in this paper differs from the above mentioned approaches in a number of ways. We use neither a map, nor localization in a shared frame of reference. In our system robots deploy a set of communication beacons into the environment in order to coordinate their motions to improve dynamic coverage. The basic assumption is that the beacon nodes that the robot deploys into the environment are small and have a compass.

3 System Architecture

In our experiments we used the Player/Stage [8, 9] simulation engine populated with simulated Pioneer 2DX mobile robots equipped with two 180° field-of-view planar laser range finders positioned back-to-back (equivalent to a 2D omnidirectional laser range finder), color camera, vision beacons and wireless communication.

Our algorithm uses two entities: the communication beacons and the mobile robots. The task of each beacon is to recommend a preferred exploration direction for robots within its communication range. Each beacon issues only a recommendation, robots combine this advice with local sensing to make a decision about which direction to actually pursue. Each robot is equipped with a 2D laser range finder

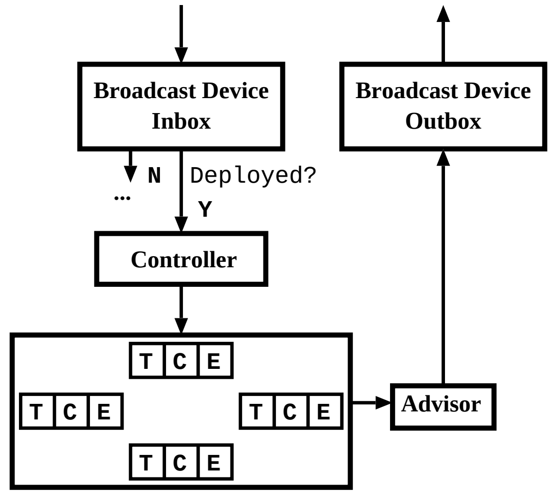

Figure 1: *Beacon Architecture*

with which it performs the coverage task. Each robot is also equipped with beacons that it deploys into the environment to help with coverage.

As shown in Figure 1, each beacon consists of a *BroadcastDevice in/outbox*, *Controller*, *States block* and *Advisor block*. If the beacon has been deployed, messages are directed to the *Controller* which parses them and updates the *States block*. The *States block* consists of four groups of states corresponding to the four directions (South, East, North, West). Every group has three states. The state T denotes whether a direction is OPEN or EXPLORED, C is a counter (if T is EXPLORED, then C counts the time since last update), and E is an extra field for network information propagation (direction of goal/home state, etc.). The *Advisor block* computes the beacon's recommendation for the best action a mobile robot should take if it is within the beacon's communication range. The computation of the recommendation is simple. All OPEN directions are recommended first (in order from South to West), followed by the EXPLORED directions with largest last update value (largest C value). The beacon's recommendation, generated by the *Advisor block*, is sent to the *BroadcastDevice outbox* for further redirection to nearby robots.

Each mobile robot is programmed using a behavior-based approach [10]. Laser, Vision and Position are the sensors being used. Position is a virtual sensor that combines odometry and compass. Arbitration [11] is used for behavior coordination. Priorities are assigned to every behavior *a priori*. As shown in Figure 2, there are five behaviors in the system: *ObstacleAvoidance*, *Repel*, *AtBeacon*, *DeployBeacon*

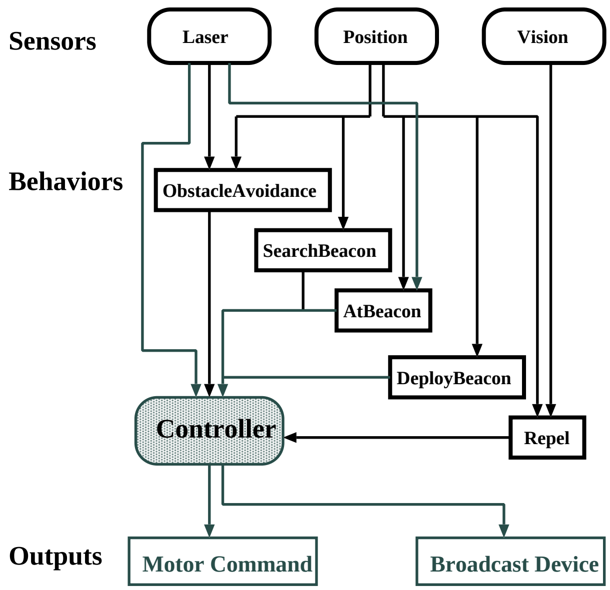

Figure 2: *System Architecture*

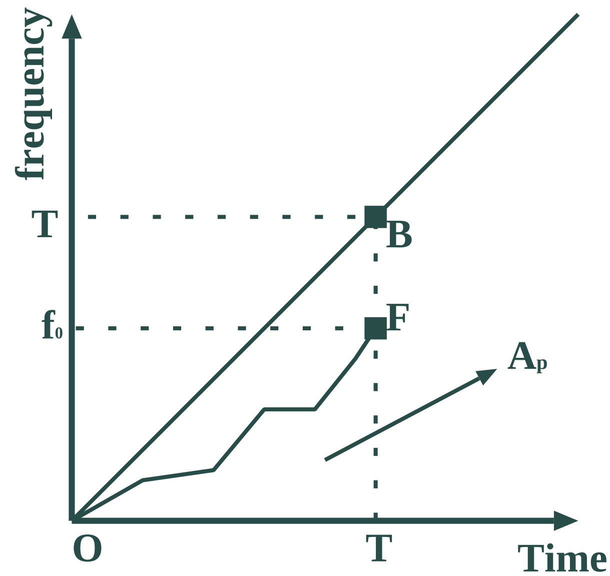

Figure 3: *Frequency Coverage of point P at time T*

and *SearchBeacon*. In addition to priority, every behavior has an activation level, which decides, given sensory input, whether the behavior should be in an active or passive state (1 or 0 respectively). Each behavior computes the product of its activation level and corresponding priority and sends the result to the Controller, which picks the maximum value, and assigns the corresponding behavior to command the Motor Controller for the next cycle. *Obstacle Avoidance* and *Repel* have the same implementation as described in [1]. One of the state variables that every robot keeps track of, is a reference to the last heard beacon. If this reference switched to a different beacon (i.e. robot moved to the communication area of a different beacon), *AtBeacon* is triggered. *AtBeacon* analyzes data messages received from the current beacon broadcasts. If the direction recommended by the beacon does not have obstacles (based on an analysis of the laser data), the robot proceeds in that direction, while sending an update beacon message (this message updates the States block in Figure 1). If, however, the suggested direction is obstructed, *AtBeacon* sends a broadcast message updating the beacon with new information and requesting a new suggestion. *SearchBeacon* is triggered after *AtBeacon* chooses and positions the robot in a certain direction. The task of *SearchBeacon* is to travel a predetermined distance. *DeployBeacon* is triggered if the robot does not receive a ”suggestion” message (i.e. a recommended direction to traverse) from any beacons. In this case the robot deploys a beacon into the environment.

4 The Frequency Coverage Metric

The idea behind the proposed metric is to compute how often points in the environment are visited (i.e. are covered by robot's sensors). Consider the graph in Figure 3. For a particular point P, which is not always under sensor coverage of any of robots, the graph of coverage frequency at time T is represented with curve OF. The ratio of area under OF to the theoretical maximum coverage (area under OB) represents the frequency coverage metric for point P at time T. Therefore, the frequency coverage metric g for all points in the environment is given by :

$$g = \frac{1}{A \cdot W} \int \int w_P \frac{2A_P}{T^2} dxdy, \qquad (1)$$

$$W = \max_P (w_P),$$

$$w_P = \begin{cases} {t_{slc}}^P & \text{if } P \text{ has been just discovered,} \\ 0 & \text{otherwise.} \end{cases}$$

where A is the area of the environment, T is the current time, A_P is a coverage frequency area of point P, ${t_{slc}}^P$ is time elapsed since last coverage of P and w_P is the weight associated with point P. Note that in order to provide a fair comparison between the previously developed static algorithm and the proposed exploratory algorithm, our present choice of w_P does not “punish” algorithms for not covering other points

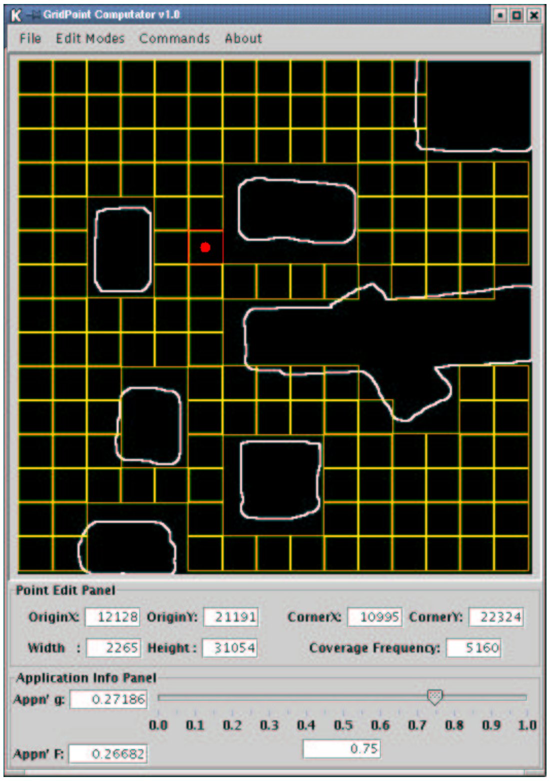

Figure 4: *Frequency Metric Estimation tool*

of the environment. The frequency coverage metric over the time of execution of the algorithm can be defined with equation 2:

$$F = \frac{1}{T}\int_0^T g \; dt \qquad (2)$$

A software application has been created which keeps track of the metric F and allows the user to define the granularity of computation, regions of coverage interest, and shows coverage statistics about points in the environment. A screen shot of the application is shown in Figure 4. The essential principle of the tool is that for every point it determines whether the point is in sensor shadow of any of the robots or not and update the statistics appropriately. Note that F is a cumulative metric. In our experiments F converged to a steady state. Therefore, a single experiment that results in that state is considered enough to establish the value of F for a given configuration of robots in a team.

5 Approach

The fundamental idea of the algorithm is simple. A robot explores as long as there are OPEN regions left.

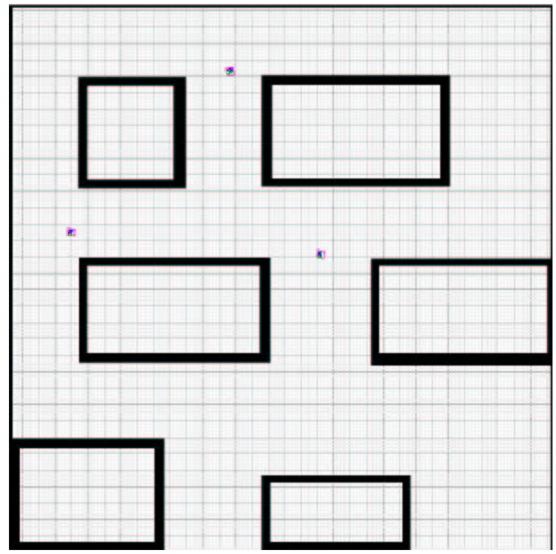

Figure 5: *The simulation environment*

If all regions are EXPLORED, then the robot picks the direction which was least recently explored. As discussed in Section 3, decisions of which direction to explore next are made by beacons. The robots, however, may alter those decisions if real world observations (through laser data analysis) diverge from the beliefs of beacons. In addition, the Repel behavior ensures that the robots spread out in the environment as far as possible so that the coverage is more effective. Values for communication radius of beacons and the minimal distance between the beacons were picked empirically.

6 Experiments and Results

We experimentally tested the proposed approach and compared its performance using the frequency coverage metric with the *Molecular* approach described in [1].

Briefly put, the *Molecular* approach causes robots to disperse as far as possible into the environment. Every robot selects a 'repelled' direction of motion, which is diametrically opposite to the average angle subtended by all its neighbors in its visual field. A set of local algorithms maximizing sensor coverage are then applied. [1] shows that *Molecular* solves a particular coverage problem within 5-7% of the manually derived optimal case.

Note that the *Molecular* approach is a static deployment algorithm. Therefore, for the purpose of reasonable comparison, the conditions of task termination of *Molecular* were relaxed, so that after set-

tling down into the equilibrium state the motion of robots does not stop and its frequency coverage metric would not deteriorate. The general setup of the experiments for both approaches consisted of teams of size 1, 2, 3, 4 and 7 robots. Figure 5 shows the experimental environment. A trial terminates when F achieves a steady state. In other words, if there is no $\varepsilon = 10^{-5}$ change in F for a certain amount of time, F is considered to have reached a steady state. As shown in Figure 6, the proposed

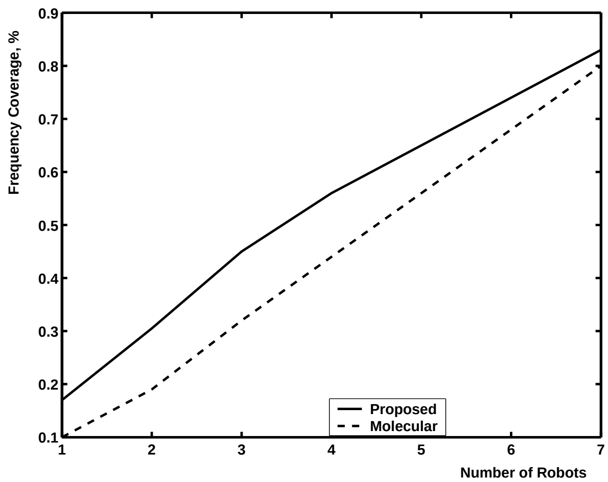

Figure 6: *Results of experimental simulations for groups of robots for the Beacon-based and Molecular algorithms*

technique outperforms the *Molecular* approach. In addition to a better dynamic coverage result, the proposed algorithm deploys a static beacon network in each of the test cases, that can be used for a number of applications later. The comparison is summarized in Table 6.

Table 1: *Experimental Data*

Team Size	Molecular	Beacon-based
1	0.1%	0.17%
2	0.19%	0.305%
3	0.32%	0.45%
4	0.44%	0.56%
7	0.8%	0.81%

7 Discussion

As seen from the results presented in Figure 6, the difference in performance between the two approaches decreases with increase in the number of robots. This is to be expected however, since as the environment becomes saturated with robots, static algorithms deploy robots such that they cover the whole environment without significant motion at all times. Therefore, the cases of interest here are those of group sizes 1, 2 and 3 robots. Note that the environment is of a rather small size, compared to the range of laser sensors, and therefore, the environment becomes saturated even with 4 robots. As shown in Figure 7, the successfully deployed static beacon network shows that the proposed algorithm forces

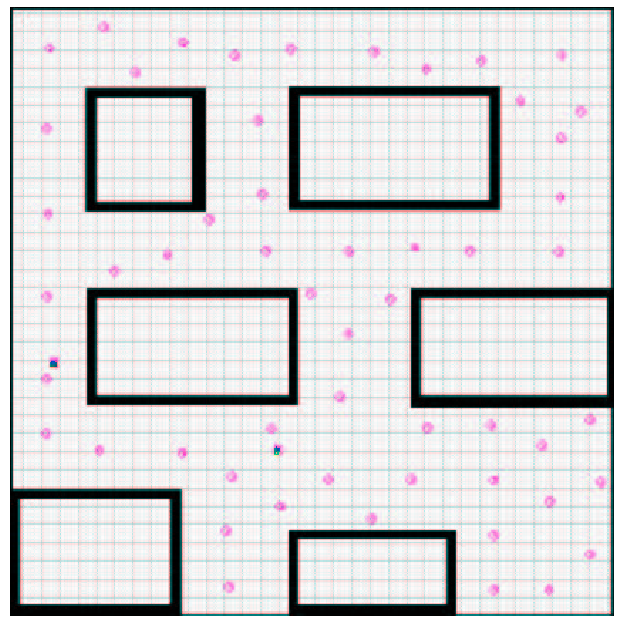

Figure 7: *Final state of the deployed static sensor network for the team of 2 robots*

robots to explore every region of the environment. Moreover, a patrolling behavior emerges where robots tend to revisit points over time, since they are not allowed to settle. Two parameters that were picked arbitrarily for the proposed algorithm are the beacon communication radius R_b and distance that the robot traverses between two neighboring beacons D_b (this is not required to be precise and our simulations trials incorporate some odometric uncertainty in the form of contaminant noise). In our present implementation $R_b = 3m < D_b = 4.5m$, where D_b varies due to noise. Note also that the deployed static beacon network does not represent an *ideal* grid, but rather, due to additive noise in the simulation, is distorted. However, the performance of the algorithm does not deteriorate, showing robustness and reliability as some of its primary qualities.

8 Conclusion and Future Work

We introduced the dynamic coverage task and proposed a decentralized solution to it. Our solution to the problem relies on the deployment of beacons into

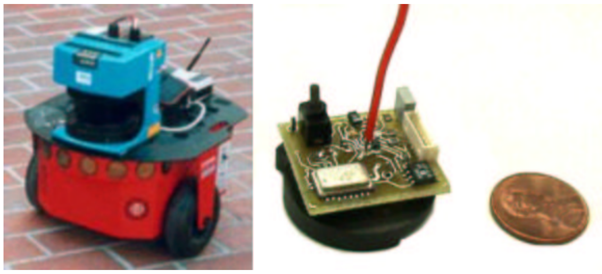

Figure 8: *Robotic platform for exploration experiments.* **(left)** *Pioneer 2DX robot, the carrier.* **(right)** *A mote, several of which would be used as beacons.*

the environment as support infrastructure which the robots use to solve the coverage problem. Robots explore the environment, and based on certain local criteria, drop a beacon into the environment, from time to time. We compared proposed algorithm to a (modified) version of a prior approach in which robots disperse themselves in the environment by locally repelling interaction in order to 'fan out' to expand coverage. Simulations show that proposed algorithm outperforms *Molecular* algorithm. In addition, the present algorithm deploys a static network of nodes into the environment which has applications other than coverage.

In our future work we plan to extend the weights w_P so that it would punish the coverage algorithm for not visiting areas of the environment. We also plan to investigate the possibility of assigning priority-areas *a priori* for frequent coverage. As pointed out in section 5, the values for each beacon's communication radius and the inter-beacon distance were picked in an adhoc manner. We plan to investigate the optimal dependencies between the two values, their effect on the coverage frequency metric F and the number of beacons required for a given environment. The number of beacons available to a robot is assumed to be infinite in this paper (even though in practice of course only a finite number are needed/used). We plan to extend our algorithm with switching from exploration mode (search for OPEN directions) to patrolling mode (follow the EXPLORED directions) in case the robot runs out of beacons. We also plan to conduct physical robot experiments with several Pioneer 2DX robots carrying communication beacons (motes) onboard and accomplishing the frequency coverage task. Figure 8 shows the physical platform for our experiments.

In future work we also plan to exploit the deployed static sensor network for other behaviors. One example is recovery, when after being deployed, every robot uses the network to return to "home base". The propagation of information through the network could also dramatically increase performance of the coverage algorithm itself (e.g. by dynamically adjusting the beacon drop-off distance).

Acknowledgments

This work is supported in part by DARPA grant DABT63-99-1-0015 and NSF grants ANI-9979457 and ANI-0082498.

References

[1] M. A. Batalin and G. S. Sukhatme, "Spreading out: A local approach to multi-robot coverage," in *Proc. of 6th International Symposium on Distributed Autonomous Robotic Systems*, Fukuoka, Japan, 2002.

[2] D. W. Gage, "Command control for many-robot systems," in *the Nineteenth Annual AUVS Technical Symposium*, Huntsville, Alabama, USA, 1992, pp. 22–24.

[3] B. Yamauchi, "Frontier-based approach for autonomous exploration," in *In Proceedings of the IEEE International Symposium on Computational Intelligence, Robotics and Automation*, 1997, pp. 146–151.

[4] B. Yamauchi, A. Schultz, and W. Adams, "Mobile robot exploration and map-building with continuous localization," in *In Proceedings of the 1998 IEEE/RSJ International Conference on Robotics and Automation*, 1998, vol. 4, pp. 3175–3720.

[5] A. Zelinsky, "A mobile robot exploration algorithm," in *IEEE Transactions on Robotics and Automation*, 1992, vol. 8, pp. 707–717.

[6] W. Burgard, D. Fox, M. Moors, R. Simmons, and S. Thrun, "Collaborative multirobot exploration," in *Proc. IEEE ICRA*, 2000.

[7] A. F. Winfield, *Distributed sensing and data collection via broken ad hoc wireless connected networks of mobile robots*, vol. 4, pp. 273–282, Springer, 2000.

[8] B. P. Gerkey, R.T. Vaughan, K. Stoy, A. Howard, G.S. Sukhatme, and M.J. Mataric, "Most valuable player: A robot device server for distributed control," in *IEEE/RSJ Intl. Conf. On Intelligent Robots and Systems (IROS)*, Wailea, Hawaii, 2001.

[9] R.T. Vaughan, "Stage: a multiple robot simulator," Tech. Rep. IRIS-00-393, Institute for Robotics and Intelligent Systems, University of Southern California, 2000.

[10] M. J. Mataric, "Behavior-based control: Examples from navigation, learning, and group behavior," *Journal of Experimental and Theoretical Artificial Intelligence, special issue on Software Architectures for Physical Agents*, vol. 9, no. 2-3, pp. 323–336, 1997.

[11] P. Pirjanian, *Behavior Coordination Mechanisms - State-of-the-art*, Ph.D. thesis, University of Southern California, October 1999.